

Wolfram | Alpha не е нито Google, нито IBM Watson, нито Wikipedia.



Enter what you want to calculate or know about

Extended Keyboard

Upload

Examples

Random

Compute expert-level answers using Wolfram's breakthrough algorithms, knowledgebase and AI technology

Mathematics ›

Science & Technology ›

Society & Culture ›

Everyday Life ›



Step-by-Step Solutions



Units & Measures



People



Personal Health



Elementary Math



Physics



Arts & Media



Personal Finance



Algebra



Chemistry



Dates & Times

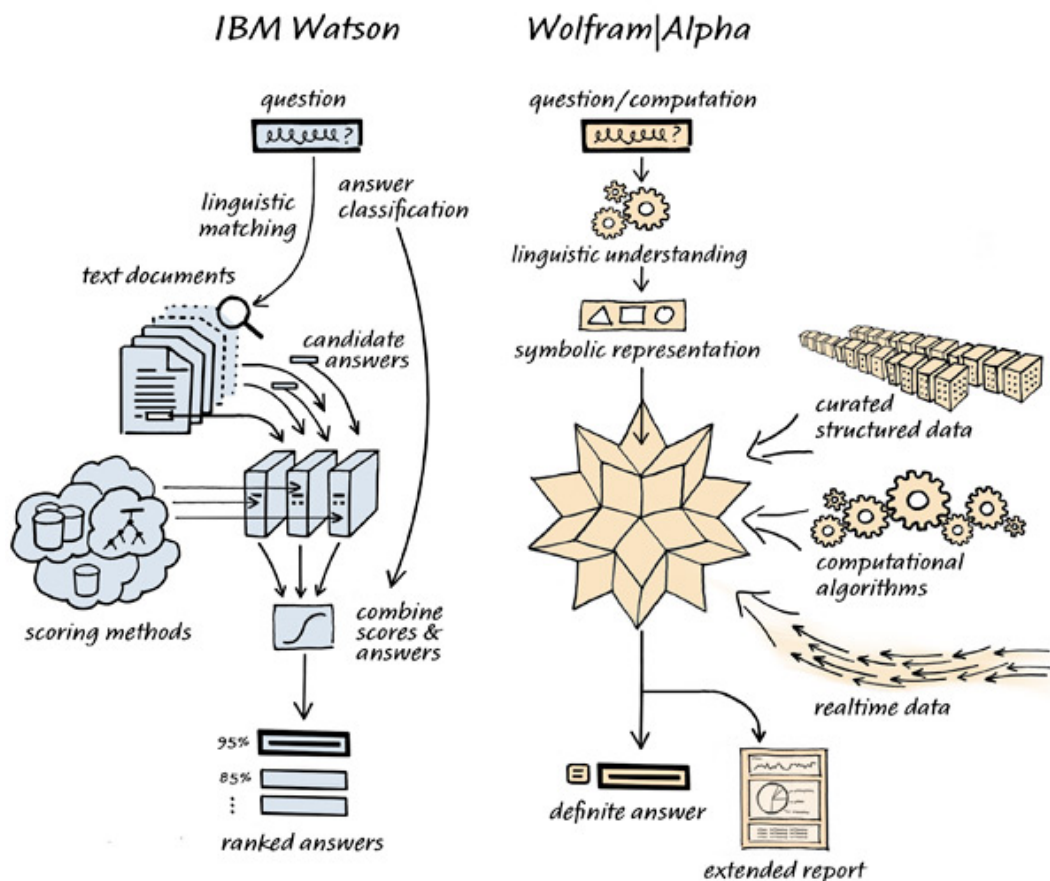


Surprises

фиг 1.1

С тази огромна система на *Wolfram Research* по блестящ начин показват своята работа - и като цяло възможностите им, които са доста завидни. Каквото и да се случи като промяна, системата на IBM със сигурност би трябвало да успее да спечели, но едно е ясно: това, което *Wolfram Research* направи, ще има важен ефект в промяната на очакванията на хората за това как те биха могли да взаимодействат с компютрите.

фиг 1.2



Когато стартира *Wolfram|Alpha*, първоначално хората продължиха да го наричат като „нова търсачка“ - тъй като всъщност **търсенето по ключови думи (keyword-search)** беше единственият модел, който имаха, за да намерят информация в голям мащаб. Но проектът на *Wolfram Research* дава страхотен пример за друг модел: **отговор на въпроси (question-answering)**. И когато хората придобият субективна представа за този модел, те ще се приближават много по-близо до осъзнаването на възможностите с това, което са изградили във *Wolfram|Alpha*.

На диаграмата на *фиг. 1.2*, освен, че показва каква е релацията и разликата между *Wolfram|Alpha* и проекта на *IBM Jeopardy*, за който ние няма да задълбаваме много сега (но е доста интересна тема, ако някой друг реши да потърси информация за нея) демонстрира и обща картина на структурата и архитектурата на тази система.

Основният подход на IBM има дълга история с редица в областта на извличане на информация примери, която по много начини се споделя с търсачките. Съществена идея е да се започне с текстови документи и след това да се изгради система за статистическо съвпадение на въпроси, които се задават на отговори, представени в документите. (Първата стъпка е да се търсят текстови съвпадения на въпрос - използване на лингвистични модели на сравнения. По-трудната работа е след това да вземете списъка с потенциални отговори и да използват разнообразие от различни методи, за да се оценят, и накрая да се комбинират тези резултати, за да се избере най-добър отговор.)

Ранните версии на този подход се връщат близо 50 години назад, до първата фаза на изследванията за изкуствен интелект. Постигнат е постепенно напредък - по-специално, както е проследен през последните 20 години в годишния TREC (Text Retrieval Conference) въпрос, отговарящ на конкуренцията.

***Wolfram | Alpha* е съвсем различен вид нещо - нещо много по-радикално, основано на съвсем различна парадигма.** Ключовият момент е, че *Wolfram|Alpha* не се занимава с документи или нещо получено от тях. Вместо това тя се занимава директно със сурови, прецизни, изчислими знания. И това, което е вътре в него, не е статистическо представяне на текст, а реално представяне на знанието.

Въвеждането на *Wolfram | Alpha* може да бъде въпрос на естествен език. Но това, което *Wolfram|Alpha* прави, е да преобразува този естествен език в прецизна изчислима вътрешна форма. И тогава той приема тази форма и използва изчислимите си знания, за да изчисли отговор на въпроса (Интересен факт: част от разработчиците на тази система споделят, че първоначално изобщо не са знаели и не са били сигурни дали това което са започнали да разработват като идея е възможно да се случи).

В известен смисъл *Wolfram|Alpha* напълно разбира всеки отговор, който дава. Това не е някакво предаване на парчета статистически съвпадения на документи, които са били подадени. Всъщност изчислява отговорите си въз основа на знанията, които има. И повечето отговори, които изчисляват, са напълно НОВИ: никога не са били изчислявани или записвани преди.

Във *Wolfram|Alpha* по-голямата част от работата е просто добавяне на изчислими знания към системата. Събиране на данни, свързване на емисии в реално време, инжектиране на специфични за домейна експертизи, прилагане на изчислителни алгоритми - и изграждане на вид обобщена граматика, която улавя естествения език, използван за заявки. При разработването на системата, разработчиците постоянно изграждат различни области на знанието, съсредоточавайки се първо върху тези, които адресират доста кратки въпроси, които хората задават, и които са важни на практика. Те са почти точно в противоположния край на нещата от необходимото в *IBM Jeopardy* системата - и от директния път, който IBM е поел към тази цел. Няма съмнение, че след време *Wolfram|Alpha* ще успее да направи неща като задачата на *IBM Jeopardy* - макар и по съвсем различен начин от системата на IBM - но това не е нещото за което е създадена системата.

(Интересна метрика е, че *Wolfram|Alpha* в момента знае(т.е. може да отговори) за три четвърти от субектите, които възникват при въпроси задавани в *IBM Jeopardy* – което като се вземе предвид, че *Wolfram|Alpha* изобщо не е създадена с тази цел си е нещо доста силно и може да се разглежда и като огромно предизвикателство към гиганта IBM.)

Какво още имаме тук:

Просто API; за бързо разпознаване на заявки; за кратки отговори; на резюмета; за гласов отговор; за незабавно калкулиране; за пълен отговор; за постъпково показване на решение; за разговор (добавяне на интелигентност към вашия бот с резултати, оптимизирани за диалог със системата) и т.н и т.н.

Причината за популярност на системата е наистина доста проста: *Wolfram|Alpha* **напълно променя икономиката на много от програмирането като цяло**. В днешни дни забележителен брой програми разчитат на това, че имат някакъв вид знания. И традиционно, единственият начин да вкарате знания в дадена програма беше програмистът старателно да я постави там. Но с *Wolfram|Alpha* в картинката, това е различна история. Защото вграденият във *Wolfram|Alpha* интелект е огромен по обем и изчислими знания. **И ако дадена програма е свързана с *Wolfram | Alpha*, тя може незабавно да използва цялото това знание.**

Независимо дали човек изгражда уебсайт или мобилно приложение, софтуер за настолни компютри или корпоративно приложение, въпросът е, че човек може да използва *Wolfram|Alpha* като платформа за „изчисления, базирани на знанието“, така че притежаването на всякакъв вид изчислими знания става ефективно без инженерна гледна точка.

API Explorer

API Documentation »

Enter a query

Select formats for XML output

☒ Images ☒ Plaintext ☐ Mathematica Input
☐ Sound ☐ HTML ☐ Mathematica Cells

Submit

Web version of this result »

<http://api.wolframalpha.com/v2/query?appid=xxxx&input=2+2&format=image,plaintext>

API response

```
<?xml version="1.0" encoding="UTF-8"?>
<queryresult success="true"
  error="false"
  sumcode="5"
  datatypes="Math"
  timeout=""
  timing="0.24"
  parsetiming="0.054"
  parsetimeout="false"
  recalculate=""
  version="2.0">
  <pod title="Input"
    scanner="Identity"
```

Е, ако бяхте бяхте програма ☺, всичко, което бихте направили, е - вместо това да използвате URI във формата <http://api.wolframalpha.com/...> И това, което бихте получили обратно, е структура, която бихте могли да използвате.

Подобно на повечето съвременни API, *Wolfram|Alpha* API е така нареченият API в стил **REST**, който връща **XML изход**. Системата може също да има пълно символично представяне на всяко парче от изхода, дадено като символен израз на *Mathematica*. Излишно е да казвам, че има всякакъв вид подробни контроли. За определяне на геолокацията да предположим. За уточняване дали резултатите трябва да се връщат синхронно всички заедно или асинхронно на парчета, когато са готови. И така нататък. Има и цял разширен език за *Wolfram|Alpha* заявки, който дава възможност например да се уточни кои конкретни изходни парченца трябва да се генерират или какви предположения трябва да се правят за видовете обекти във входа.

В крайна сметка всички тези възможности могат да бъдат достъпни само чрез изпращане на съответния низ до сървърите *Wolfram|Alpha*. Но за удобство вече съществуват библиотеки за много общи езици (. Net, Perl, Python, Ruby, PHP и Java, да не говорим за *Mathematica*), които настройват низа на заявката и след това манипулират получената структура, която се връща.

Има много софтуерни проекти, които вече използват API на *Wolfram|Alpha*. Един забележителен пример е *Bing* в *Microsoft*, където заявките се изпращат от търсачката *Bing* до *Wolfram|Alpha API*. Друг пример с доста различен характер е в електронните книги на *Touch Press*, където *Wolfram|Alpha* се използва за генериране на динамични елементи в книгите.

Качествени характеристики като функционалност, надеждност, използваемост и ефективност са оказали огромно влияние по време на разработката на системата.

Достоверността на отговора на системата както уточнихме не се базира на статистики, нито да данни търсени в мрежата, а на значително количество изкуствен интелект натрупан в нея, който трябва да се синхронизира с много фактори за да бъде надежден като възможност за получаване на отговор. Описаните множества интерфейси и начина, по който се подава съответния отговор от системата е повлиян значително от използваемостта на приложението и желанието на разработчиците да го направят възможно най-полезен и лесно използваем не само от крайния потребител а и от други приложения. Ефективността е задължила използването на доста гъвкави, умни и бързи алгоритми при пресмятането на точните данни желани от потребителя.

Източници:

- 1.) <https://www.wolframalpha.com/about/>
- 2.) <https://writings.stephenwolfram.com/2011/01/jeopardy-ibm-and-wolframalpha/>
- 3.) <https://writings.stephenwolfram.com/2011/01/knowledge-based-computing-and-version-2-0-of-the-wolframalpha-api/>
- 4.) <https://www.linkedin.com/pulse/wolframalpha-google-ibm-watson-wikipedia-john-cassel>
- 5.) <https://www.zdnet.com/article/wolfram-alpha-vs-ibms-watson-how-they-think/>
- 6.) <https://products.wolframalpha.com/api/>
- 7.) <https://www.wolfram.com/mathematica/>

Андрей Стоев, ф.н. 62369, СИ 3-та група